

City University of New York (CUNY) CUNY Academic Works

Open Educational Resources

City College of New York

Fall 2018

CSC 59939 Topics in Modern Software Engineering

Nikolai Avteniev
CUNY City College

NYC Tech-in-Residence Corps

[How does access to this work benefit you? Let us know!](#)

Follow this and additional works at: https://academicworks.cuny.edu/cc_oers

 Part of the [Computer Sciences Commons](#)

Recommended Citation

Avteniev, Nikolai and NYC Tech-in-Residence Corps, "CSC 59939 Topics in Modern Software Engineering" (2018). *CUNY Academic Works*.
https://academicworks.cuny.edu/cc_oers/128

This Syllabus is brought to you for free and open access by the City College of New York at CUNY Academic Works. It has been accepted for inclusion in Open Educational Resources by an authorized administrator of CUNY Academic Works. For more information, please contact AcademicWorks@cuny.edu.

Topics in Modern Software Engineering

Instructor

- Nikolai Avteniev
- navteniev@linkedin.com
- <https://www.linkedin.com/in/navteniev/> on LinkedIn

Textbook, title, author, and year

- Agile Software Engineering, Orit Hazzan, Yael Dubinsky, Springer Science & Business Media, 2009, ISBN-10: 1848001991
- Other materials will be sourced from published empirical research, and content from software engineering blog posts.
- Eg. Continuous Deployment of Mobile Software at Facebook (Showcase) ACM SIGSOFT: International Symposium on the Foundations of Software Engineering (FSE 2016)

Specific course information

- The iterative and incremental software development life cycle from idea to working software. Documentation of emergent design and architecture. Documentation of software development process. Software development process iteration. Written report and oral presentation on selected topics in software engineering. Written reports and oral presentation of projects.

Specific goals for this course

- 1) Students will acquire knowledge of Agile software development
- 2) Students will acquire knowledge of automated testing techniques
- 3) Students will acquire knowledge of continuous integration and continuous deployment techniques
- 4) Students will acquire knowledge of collaborative planning and estimating
- 5) Students will acquire knowledge of collaborative implementation techniques
- 6) Students will acquire knowledge of agile software architecture techniques
- 7) Students will acquire ability to work effectively as a part of an agile team
- 8) Students will acquire ability to design and build large systems in iterative increments
- 9) Students will acquire knowledge of the business responsibilities of the software engineer
- 10) Students will acquire knowledge of empirical software engineering practices
- 11) Students will acquire knowledge of industrial software engineering practices
- 12) Students will acquire knowledge of User Experience Design UX

Grading

- 60% term project
 - 50% individual contributions
 - Code quality, testing quality, estimation accuracy, documentation, class room presentation

- 50% collaborative work
 - Working session participation, Pull Request Feedback, Task allocation, Incremental progress, Pair programming
- Both the developed software product and the established software development process will be evaluated.
- 20% research paper
 - A paper on a topic in software engineering, eg. Effectiveness of Code Reviews, Effectiveness of Test Driven Development, Effectiveness of Pair Programming
 - The paper will be accompanied by an oral presentation.
- 20% class activity participation
 - Class attendance is mandatory, each session will include a group exercise
 - Taking part in group exercise will be mandatory in order to achieve full credit

Brief list of topics to be covered

Week	Topics
1	Introduction to Agile Software Development
2	Software Development is a Team Sport
3	Role of Customers and Users in Software Development
4	Planning and Estimating
5	Measuring Software Development, What and How?
6	Building Quality in Role of Test Automation in Software Development, Perpetual Beta, Continuous Integration and Delivery
7	Battling Complexity, Source Code Management, Continuous Refactoring, and Peer Code Review
8	Role of Architecture and Design in Agile Software Development
9	Iterating on Agile, Continuous Improvement
10	Seeing the Code, Monitoring and Observing your Systems in Production
11	Infrastructure as a Service, Agile Infrastructure on the Public Cloud
12	When Things Go Wrong, Handling Production Issues
13	Making Users Awesome, UX / HCI
14	Modern Agile, where do we go from here?